USSN 10/613,750

PATENTS • DESIGNS • The Patent Office • COPYRIGHT • TRADE MARKS

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed AnBrewer.

Dated 28 August 2002

Patents Form 1/77

Patents Act 1977
(Rule

USSN 10/613,750

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

The Patent Office

Cardiff Road
Newport
South Wales
NP10 8QQ

1. Your reference

300201957-01 GB

2. Patent application number
(The Patent Office will fill in this part)

0215524.0

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

Hewlett-Packard Company
3000 Hanover Street
Palo Alto
CA 94304, USA

Patents ADP number *(if you know it)*

If the applicant is a corporate body, give the country/state of its incorporation

Delaware, USA

49658004

4. Title of the invention

Method And Apparatus For Generating
A Cryptographic Key

5. Name of your agent *(if you have one)*

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

Chris Harrison
Hewlett-Packard Ltd, IP Section
Filton Road, Stoke Gifford
Bristol BS34 8QZ

Patents ADP number *(if you know it)* 8191439001

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country | Priority application number *(if you know it)* | Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application | Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

Yes

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	18
Claim*(s)*	5
Abstract	1
Drawing*(s)*	2 +2

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	1
Request for preliminary examination and search *(Patents Form 9/77)*	1
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	Fee Sheet

11. I/We request the grant of a patent on the basis of this application.

Signature

Date 4/ 7/2002

12. Name and daytime telephone number of person to contact in the United Kingdom

Tony Judd Tel: 0117-312-8026

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) If you need help to fill in this form or you have any questions, please contact the Patent Office on 08459 500505.

b) Write your answers in capital letters using black ink or you may type them.

c) If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.

d) If you have answered 'Yes' Patents Form 7/77 will need to be filed.

e) Once you have filled in the form you must remember to sign and date it.

f) For details of the fee and ways to pay please contact the Patent Office.

METHOD AND APPARATUS FOR GENERATING A CRYPTOGRAPHIC KEY

The present invention relates to a method and apparatus for generating a cryptographic key.

A key feature associated with cryptography is the provision of a trust authority, where a trust authority is responsible for issuing private and public keys to appropriate individuals/entities. However, as a private key, is by its nature, private to a specific individual/entity it is essential that a user can trust that the trust authority will not disclose or otherwise use the user's private key in an inappropriate manner. However, it can be difficult for a user to build a strong trust relationship with a single trust authority.

One solution to this problem has involved the use of a plurality of trust authorities to generate individual parts of a private key, where no one trust authority has access to the complete private key. In particular, one solution involves the use of a shared secret in which a group of trust authorities use the shared secret to generate their portion of the private key. However, this solution requires the use of a trusted secret distributor.

Another solution involves each trust authority providing a portion of a private key based upon the identity of the user where the identity of the user is the same for each trust authority. However, in many applications a user may have different identities when dealing with the different trust authorities.

It is desirable to improve this situation.

In accordance with a first aspect of the present invention there is provided a computer apparatus comprising a processor arranged to generating a cryptographic key using a first data set that corresponds to a first identifier, a second data set that corresponds to a first trusted party's public key, a third data set that corresponds to a second identifier and a fourth data set corresponds to a second trusted party's public key.

Preferably the first data set is a first public parameter.

Suitably the second data set is a second public parameter.

Suitably the first data set is a first private parameter generated by the first trusted party.

Preferably the third data set is a third public parameter.

Suitably the fourth data set is a fourth public parameter.

Suitably the third data set is a third private parameter generated by the second trusted party.

Suitably the cryptographic key is an encryption key.

Suitably the processor is arranged to encrypt a fifth data set with the encryption key.

Suitably the processor is arranged to encrypt the fifth data set with the encryption key and a random number.

Suitably the processor is arranged encrypt the fifth data set using a bilinear pairing, such as a Tate or Weil pairing, when operating on the first and second data sets and the third and fourth data sets.

Suitably the cryptographic key is a decryption key.

Suitably the processor is arranged to decrypt an encrypted data set with the decryption key.

Suitably the processor is arranged decrypt the encrypted data set using a bilinear pairing, such as a Tate or Weil pairing, when operating on the first and second data sets and the third and fourth data sets.

Suitably the cryptographic key is a signature key.

Suitably the processor is arranged to sign a sixth data set with the signature key.

Suitably the processor is arranged to sign the sixth data set with the signature key and a random number.

Suitably the processor is arranged to sign the sixth data set using a bilinear pairing, such as a Tate or Weil pairing, when operating on the first and second data sets and the third and fourth data sets.

Suitably the cryptographic key is a verification key.

Suitably the processor is arranged to verify a signed data set with the verification key.

Suitably the processor is arranged to verify the signed data using a bilinear pairing, such as a Tate or Weil pairing, when operating on the first and second data sets and the third and fourth data sets.

In accordance with a second aspect of the present invention there is provided a method comprising generating a cryptographic key using a first data set that corresponds to a first identifier, a second data set that corresponds to a first trusted party's public key, a third data set that corresponds to a second identifier and a fourth data set that corresponds to a second trusted party's public key.

Preferably the method further comprises encrypting a fifth data set with the cryptographic key.

Preferably the fifth data set is encrypted using a bilinear pairing, such as a Tate or Weil pairing, when operating on the first and second data sets and the third and fourth data sets.

In accordance with a third aspect of the present invention there is provided a computer system comprising a first computer entity arranged to generate a first data set that corresponds to a first trusted party's public key; a second computer entity arranged to generate a second data set that corresponds to a second trusted party's public key; and a third computer entity arranged to generate a cryptographic key using a first identifier in conjunction with the first data set and a second identifier in conjunction with the second data set.

Preferably the third computer entity is arranged to encrypt a third data set with the cryptographic key.

Preferably the third computer entity encrypts the third data set using a bilinear pairing, such as a Tate or Weil pairing, when operating on the first and third data sets and the second and fourth data sets.

Preferably the first data set and second data set are public data parameters.

Preferably the public data parameters include an elliptic curve and a generator point on the elliptic curve.

For a better understanding of the present invention and to understand how the same may be brought into effect reference will now be made, by way of example only, to the accompanying drawings, in which:-

Figure 1 illustrates a computer system according to an embodiment of the present invention;

Figure 2 illustrates a computer system according to an embodiment of the present invention.

Figure 1 shows a first computer entity 10, a second computer entity 20, a third computer entity 30 and a fourth computer entity 40 connected via a network 50, for example the Internet.

The first computer entity 10 represents a first trust authority 60, for example a company, the second computer entity 20 represents a second trust authority 70, for example a division within the company and the third computer entity 30 represents a user 80, for example a worker within the company. The fourth computer entity 40 represents, for example, a business partner 90 of the company that wishes to interact with the user 80.

The first, second, third and fourth computer entities 10, 20, 30, 40 are conventional computing devices as is well known to a person skilled in the art.

The first computer entity 10 and second computer entity 20 form a trust authority hierarchy in which the first computer entity 10 acts as a root trust authority and the second computer entity 20 acts as a middle level trust authority, thereby forming a public-key infrastructure. As described in detail below, on receipt by the second computer entity 20 of a master private key generated by the first computer entity 10 the second computer entity 20 is able, using identifier-based cryptography, to generate a private/public key pair without further interaction from the first computer entity 10, where the public key can be verified, without the need for digital certificates, such that the verifier can be convinced that the public key could only be generated with knowledge of the master private key generated by the first computer entity 10.

The following embodiment utilises identifier-based cryptography using Tate pairing to provide multiple levels of trust authorities, however other types of pairing may also be used, for example Weil pairings.

For the purposes of this embodiment *G1* and *G2* denote two groups of prime order q in which the discrete logarithm problem is believed to be hard and for which there exists a computable bilinear map, for example, a Tate pairing.

i.e. $t : G_1 \times G_1 \longrightarrow G_2$

G_1 is a group of points on an elliptic curve and G_2 is a subgroup of a multiplicative group of a finite field.

As the mapping between G_1 and G_2 is bilinear exponents/multipliers can be moved around. For example if $a, b, c \in \mathbb{F}_q$ and $P, Q \in G_1$ then

$$\begin{aligned} t(aP, bQ)^c &= t(aP, cQ)^b = t(bP, cQ)^a = t(bP, aQ)^c = t(cP, aQ)^b = t(cP, bQ)^a \\ &= (abP, Q)^c = t(abP, cQ) = t(P, abQ)^c = t(cP, abQ) \\ &= \ldots \\ &= t(abcP, Q) = t(P, abcQ) = t(P, Q)^{abc} \end{aligned}$$

Additionally, for the purposes of this embodiment the following cryptographic hash functions are defined:

$$H_1 : \{0,1\}^* \longrightarrow G_1$$
$$H_2 : \{0,1\}^* \longrightarrow \mathbb{F}_q$$
$$H_3 : G_2 \longrightarrow \{0,1\}^*$$

To provide a trust hierarchy a public/private key pair is defined for a trust authority where the public key R is: $R \in$ G1 and the private key s is: $s \in \mathbb{F}_q$ with R=sP where P, a public parameter, is: $P \in G_1$.

Additionally, an identifier based public key Q_{ID} / private key S_{ID} pair is defined where the Q_{ID}, $S_{ID} \in G_1$ where the trust authority's public/private key pair (R_{TA},s) is linked with the identifier based public/private key by

$$S_{ID} = sQ_{ID} \text{ and } Q_{ID} = H_1\,(\text{ID})$$

where ID is an identifier string.

Accordingly, to allow a holder of the private part s of the trust authority's public/private key pair to sign a bit string, where m denotes the message to be signed it is necessary to compute $V = sH_1(m)$. Verification requires that the following equation is satisfied:

$$t(P, V) = t(R, H_1(m))$$

This is based upon the mapping between G_1 and G_2 being bilinear exponents/multipliers, as described above. That is to say,

$$\begin{aligned} t(P, V) &= t(P, sH_1\,(m)) \\ &= t(P, H_1\,(m))^s \end{aligned}$$

$$= t(sP, H_1 (m))$$
$$= t(R, H_1 (m))$$

In particular identifier based encryption allows the holder of the private key S_{ID} of an identifier based key pair to decrypt a message sent to them encrypted using the associated public key Q_{ID} .

The message to be encrypted is denoted by *m*.

First compute $U = rP$ where r is a random element of $\mathbb{F}_q$.

Then compute $V = m \oplus H_3(t(\mathrm{R}, rQ_{ID}))$

This results in the generation of the ciphertext *U* and *V*.

Decryption of the message is performed by computing:

$$\begin{aligned} V \oplus H3\ (t(U, S_{ID})) &= V \oplus H_3(t(rP, sQ_{ID})) \\ &= V \oplus H_3(t(P, Q_{ID})^{rs}) \\ &= V \oplus H_3(t(sP, rQ_{ID})) \\ &= V \oplus H_3(t(R, rQ_{ID})) \\ &= m \end{aligned}$$

Correspondingly identifier based signatures using Tate pairing can be implemented. For example:

First compute $r = t(P, P)^k$
where k is a random element of $\mathbb{F}_q$.
Then apply the hash function H_2 to $m \| r$ (concatenation of m and r) to obtain $h = H_2(m \| r)$.
Then compute

$$U = hS_{ID} + kP.$$

Thus generating the output U and h as the signature on the message m.

Verification of the signature can be established by computing:

$$r = t(U, P)\bullet t(Q_{\mathrm{ID}}, R)^h$$

where the signature can only be accepted if $h = H_2\,(m\|r)$.

Based upon the identifier-based cryptography described above the root trust authority (i.e. the first trust authority 60) can be linked to a pseudo master private key generated by the middle level trust authority (i.e. the second trust authority 70) such that the link can be verified without the need for any digital certificates, as will now be described.

Based upon the above nomenclature table 1 lists the standard and ID based public/private key pairs that are set up for the first trust authority 60 and the second trust authority 70 where P, a pubic parameter, is an arbitrary point on an elliptic curve.

Entity	Standard Private Key	Standard Public key	ID Based Private Key	ID Based Pubic key
First TA	s_1	$R_{\mathrm{TA1}}=s_1P$		
Second TA	s_2	$R_{\mathrm{TA2}}=s_2P$	$S_{\mathrm{TA2}}=s_1Q_{\mathrm{TA2}}$	$Q_{\mathrm{TA2}}= H_1(\mathrm{TA2})$

Table 1

The second trust authority 70 creates a pseudo-master private key selecting a random number r where $r \in \mathbb{F}_q$; the random number r is the pseudo-master private key. Once the pseudo-master key has been selected the second trust authority 70 generates the following public keys:

rs_1Q_{TA2} , rP and rQ_{TA2}

It should be noted however, that even though in the above example the second trust authority 70 has created a single pseudo-master private key the second trust authority 70 could generate any number of pseudo-master private keys.

The user 80 registers with the second trust authority 70 to obtain an associated private key for the user's public key, where the user's public key could be any form of identifier, for example the user's name 'Bob', where the public key H_1(Bob) = Q_{Bob} would map to a point on an elliptic curve defined by G_1.

On registration, the second trust authority 70 provides the user 80 with the appropriate private key, which would be a combination of the user's public key and the second trust authority's pseudo private key i.e. rQ_{Bob}.

Consequently, utilizing the Tate pairing algorithms described above it is possible to verify the 'meaning' of rsQ_{TA2}, rP and rQ_{TA2} using:

$t(rP, Q_{TA2}) = t(P, rQ_{TA2})$ and
$t(P, rsQ_{TA2}) = t(sP, rQ_{TA2})$

Further (P,sP), in the above ID-based encryption and ID-based signature algorithms, can be replaced with either (P, rP) or (Q_{TA2}, rQ_{TA2}), as well as replace $t(Q_{ID}, sP) = t(sQ_{ID}, P)$ with $t(Q_{Bob}, rP) = t(rQ_{Bob}, P)$ or $t(Q_{Bob}, rQ_{TA2}) = t(rQ_{Bob}, Q_{TA2})$.

Figure 2 illustrates the same computer network as that shown in figure 1 with the addition of a fifth computer entity 100. The fifth computer entity 100 acts as another middle level trust authority (i.e. a third trust authority 200) independent of the second computer entity 20 where the first computer entity 10 is the root trust authority for both the second computer entity 20 and the fifth computer entity 100. As with the second computer entity 20 on receipt by the fifth computer entity 100 of a master private key generated by the first computer entity 10 the fifth computer entity 100 is able to generate a private/public key pair as described above. The network 50 could include additional middle level trust authorities, however, for the purposes of this embodiment only two middle level trust authorities will be described.

As described below, the user 80 has an independent identity associated with each middle level trust authority 70, 200, where each independent identity corresponds to a public key of the user 80. Each middle level trust authority 70, 200 provides a private key corresponding to the respective user's public key, as described above. To send an encrypted message to the user 80 the business partner 90 encrypts the message with a combination of the user's public keys associated with the respective middle level trust authorities 70, 200 (i.e. the user's identities associated with the respective trust authorities) and the respective trust authority's public key. To recover the encrypted message the user 80 decrypts the message with a combination of the same trust authority's public keys and the user's corresponding private key.

To sign a message a user 80 uses each trust authority's public key in combination with the user's associated private keys. To verify the signature a verifier uses a combination of the trust authority's public key with the user's corresponding public keys.

The following embodiment utilises identifier-based cryptography using Tate pairings to allow the generation of a public key that is a combination of independent identities associated with respective middle level trust authorities 70, 200.

The second trust authority 70 has a public key R_{TA1} and a corresponding private key s_1 where $R_{TA1} = s_1P$, with P being a point on an elliptic curve, as described above.

The third trust authority 200 has a public key R_{TA2} and a corresponding private key s_2 where $R_{TA2} = s_2P$, with P being a point on an elliptic curve, as described above.

For n trust authorities the public/private key pair could be generalised by:

$$R_{TAi} = s_iP$$

Associated with each middle level trust authority 70, 200 the user 80 has a independent identity, that is to say with the second trust authority 70 the user 80 has an identity ID1, for example the user's name 'Bob', with third trust authority 200 the user 80 had another identity ID2, for example the name of the company the user 80 works for.

Accordingly, the user 80 has independent identity based private keys and public keys with each middle level trust authority 70, 200, where the user's identity based public key with the second trust authority 70 is $Q_{ID1} = H_1(\text{ID1})$ and the user's identity based private key with the second trust authority 70 is S_1, where $S_1 = s_1Q_{ID1}$ and the user's identity based public key with the third trust authority 200 is $Q_{ID2} = H_1(\text{ID2})$ and the user's identity based private key with the third trust authority 200 is S_2, where $S_2 = s_2Q_{ID2}$.

To allow the business partner 90 to encrypt a message m for the user 80 based upon the independent identities associated with each middle level trust authority 70, 200 the business partner 90 generates ciphertext V and U, where:

$$V = m \oplus H_3\left(\prod_{i=1}^{2} t(R_{TAi}, rQ_{IDi})\right)$$

and

$$U = rP$$

where r is a random number selected by the business partner 90.

Decryption is performed by computing:

$$m = V \oplus H_3\left(t(U, \sum_{i=1}^{2} S_i)\right)$$

Accordingly, message m can only be decrypted with knowledge of both private keys S_1, S_2.

The following embodiments utilises identifier-based cryptography using Weil pairings to allow the generation of a public key that is a combination of independent identities associated with respective middle level trust authorities 70, 200. In a more general case, the trusted authorities can be totally independent to each other and there is no needs for any business relationship to exist between the trust authorities, in fact the trust authorities do not need to know each other. For example the trust authorities may not belong to the same root trusted authority. Indeed, one or more of the trust authorities could be a root authority.

The first embodiment utilizing Weil pairings allows the business partner 90 to encrypt a message $m \in \{0,1\}^n$ for the user 80, which the user can decrypt if the user 80 has a number of private keys d_{IDi} (i = 1, ..., n), each respectively issued by a trust authority TA_i (i = 1, ..., n) corresponding to a public key Q_{IDi} (i = 1, ..., n).

Each trust authority chooses a large (at least 512-bits) prime p such that $p = 2$ mod 3 and $p = 6q - 1$ for some prime $q > 3$. Further, E, an elliptic curve, is defined by $y^2 = x^3 + 1$ over $\mathbb{F}_p$.

An arbitrary point on the elliptic curve is chosen, where $P \in E/\mathbb{F}_p$ of order q.

Four hash functions are defined:
H_1: $\{0,1\}^* \rightarrow \mathbb{F}_p$;
H_2: $\mathbb{F}_{p^2} \rightarrow \{0,1\}^n$ for some n;
H_3: $\{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}^*_q$,
and H_4: $\{0,1\}^n \rightarrow \{0,1\}^n$.

Each trust authority TA$_i$ ($i = 1, \ldots, n$) respectively selects a random $s_i \in \mathbb{Z}^*_q$ and set $P_{pubi} = [s_i]P$.

The user 80 registers with each respective trust authority, providing each trust authority with an appropriate independent identifier, ID$_i$ ($i = 1, \ldots, n$) $\in \{0,1\}^*$.

Each trust authority then computes an appropriate MapToPoint (H_1(ID$_i$)) = $Q_{IDi} \in E/\mathbb{F}_p$ of order q and set the user's corresponding private key d_{IDi} to be $d_{IDi} = [s_i]Q_{IDi}$.

To encrypt a message, m, the business partner 90:

Computes a MapToPoint (H_1(ID$_i$)) = Q_{IDi} ($i = 1, \ldots, n$) $\in E/\mathbb{F}_p$ of order q.
Selects a random number $\sigma \in \{0,1\}^n$.
Computes $r = H_3(\sigma, m)$, where r is a random element that ensures only someone with the appropriate private key can decrypt the message, m.
Computes $U = [r]P$.
Computes $g_{ID} = \Pi_{(1 \leq i \leq n)}\, \hat{e}(Q_{IDi}, P_{pubi}) \in \mathbb{F}_{p^2}$.
Computes $V = \sigma \oplus H_2(g_{ID^r})$.

Computes $W = m \oplus H_4(\sigma)$.
Sets the ciphertext to be $C = (U, V, W)$.

To decrypt the message, m, the user 80:

Tests $U \in E/\mathbb{F}_p$ of order q;
Computes $x = \hat{e}(\Sigma_{(1 \leq i \leq n)} d_{IDi}, U)$;
Computes $\sigma = V \oplus H_2(x)$;
Computes $m = W \oplus H_4(\sigma)$;
Computes $r = H_3(\sigma, m)$;
Checks $U = [r]P$.

The second embodiment utilizing Weil pairings allows a user 80 to sign a message, m.

The user signs a message $m \in \{0,1\}^n$ under a number of private keys d_{IDi} ($i = 1, \ldots, n$), each respectively issued by a respective trust authority, i.e. TA_i ($i = 1, \ldots, n$) corresponding to a public key Q_{IDi} ($i = 1, \ldots, n$). The business partner 90 verifies the signature by using both the user's public keys corresponding to the signing private keys and the TA_i's public keys.

As above, each trust authority choose a large (at least 512-bits) prime p such that $p = 2 \bmod 3$ and $p = 6q - 1$ for some prime $q > 3$ with E being defined by $y^2 = x^3 + 1$ over $\mathbb{F}_p$.

An arbitary point on the elliptic curve is chosen where $P \in E/\mathbb{F}_p$ of order q.

Two hash functions are chosen:
H_1: $\{0,1\}^* \rightarrow \mathbb{F}_p$;
and H_2: $\{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}^*_q$.

Each trust authority TA_i (i = 1, ..., n) respectively selects a random $s_i \in \mathbb{Z}^*_q$ and set $P_{pubi} = [s_i]P$.

The user 80 registers with each respective trust authority providing each trust authority with an appropriate independent identity i.e. ID_i (i = 1, ..., n) $\in \{0,1\}^*$.

Each trust authority then computes an appropriate MapToPoint $(H_1(ID_i)) = Q_{IDi} \in E/\mathbb{F}_p$ of order q and sets the user's private key d_{IDi} to be $d_{IDi} = [s_i]Q_{IDi}$.

To sign a message, m, the user 80:

Selects a random $z \in \{0,1\}^n$;
Computes $U = [z]P$;
Computes $h = H_2(m, U)$;
Computes $V = [h]\,\Sigma_{(1 \leq i \leq n)}\, d_{IDi} + [z]\,\Sigma_{(1 \leq i \leq n)}\, P_{pubi}$
Ships to the business partner m, U and V.

To verify the signature (m, U, V) the business partner 90:

Computes MapToPoint $(H_1(ID_i)) = Q_{IDi} \in E/\mathbb{F}_p$ of order q;
Computes $h = H_2(m, U)$;
Computes $x = \hat{e}(P, V)$;
Computes $y = \Pi_{(1 \leq i \leq n)}\, \hat{e}(P_{pubi}, [h]Q_{IDi} + U)$;
Checks $x == y$.

The third embodiment utilizing Weil pairing provides a further embodiment that allows a user 80 to sign a message.

The user 80 signs a message $m \in \{0,1\}^n$ under a number of private keys d_{IDi} (i = 1, ..., n), each respectively issued by a respective trust authority i.e. TA_i (i = 1, ..., n) corresponding to a public key Q_{IDi} (i = 1, ..., n). The business partner

90 verifies the signature by using both the user's public keys corresponding to the signing private keys and the TA$_i$'s public keys.

As above, each trust authority choose a large (at least 512-bits) prime p such that $p = 2 \bmod 3$ and $p = 6q - 1$ for some prime $q > 3$ with E being defined by $y^2 = x^3 + 1$ over $\mathbb{F}_p$.

An arbitrary point P on the elliptic curve is chosen, where $P \in E/\mathbb{F}_p$ of order q.

Two hash functions are chosen:

H_1: $\{0,1\}^* \rightarrow \mathbb{F}_p$;

and H_2: $\{0,1\}^n \times \{0,1\}^n \rightarrow \mathbb{Z}^*_q$.

Each trust authority TA$_i$ ($i = 1, \ldots, n$) respectively selects a random $s_i \in \mathbb{Z}^*_q$ and set $P_{pubi} = [s_i]P$.

The user 80 registers with each respective trust authority providing each trust authority with an appropriate independent identity i.e. ID$_i$ ($i = 1, \ldots, n$) $\in \{0,1\}^*$.

Each trust authority computes an appropriate MapToPoint $(H_1(\mathrm{ID}_i)) = Q_{\mathrm{IDi}} \in E/\mathbb{F}_p$ of order q and sets the private key d_{IDi} to be $d_{\mathrm{IDi}} = [s_i]Q_{\mathrm{IDi}}$.

To sign a message, m, the user 80:

Selects a random $k \in \{0,1\}^n$;

Computes $e = \hat{e}(\Sigma_{(1 \le i \le n)}\, d_{\mathrm{IDi}}, P)$;

Computes $r = e^k$;

Computes $h = H_2(m, r)$;

Computes $S = ([k] - [h])\, \Sigma_{(1 \le i \le n)}\, d_{\mathrm{ID}}$;

Ships to the business partner m, h and S.

Verify the signature (m, h, S) the business partner 90:

Computes MapToPoint (H_1(ID$_i$)) = $Q_{\text{IDi}} \in E/\mathbb{F}_p$ of order q;

Computes $e' = \Pi_{(1 \leq i \leq n)}\, \hat{e}(Q_{\text{IDi}}, P_{pubi})$ – may be precomputed;

Computes $r' = \hat{e}(S, P)e'^{h}$;

Checks $h == H_2(m, r')$.

CLAIMS

1. Computer apparatus comprising a processor arranged to generating a cryptographic key using a first data set that corresponds to a first identifier, a second data set that corresponds to a first trusted party's public key, a third data set that corresponds to a second identifier and a fourth data set corresponds to a second trusted party's public key.

2. Computer apparatus according to claim 1, wherein the first data set is a first public parameter.

3. Computer apparatus according to claim 1 or 2, wherein the second data set is a second public parameter.

4. Computer apparatus according to claim 1 or 2, wherein the first data set is a first private parameter generated by the first trusted party.

5. Computer apparatus according to any of claims 1 to 4, wherein the third data set is a third public parameter.

6. Computer apparatus according to any of claims 1 to 5, wherein the fourth data set is a fourth public parameter.

7. Computer apparatus according to any of claims 1 to 5, wherein the third data set is a third private parameter generated by the second trusted party.

8. Computer apparatus according to any of the preceding claims, wherein the cryptographic key is an encryption key.

9. Computer apparatus according to claim 8, wherein the processor is arranged to encrypt a fifth data set with the encryption key.

10. Computer apparatus according to claim 9, wherein the processor is arranged to encrypt the fifth data set with the encryption key and a random number.

11. Computer apparatus according to any of claims 9 to 10, wherein the processor is arranged encrypt the fifth data set using a bilinear pairing when operating on the first and second data sets and the third and fourth data sets.

12. Computer apparatus according to claim 11, wherein the bilinear pairing is either a Tate or Weil pairing.

13. Computer apparatus according to any of claims 1 to 7, wherein the cryptographic key is a decryption key.

14. Computer apparatus according to claim 13, wherein the processor is arranged to decrypt an encrypted data set with the decryption key.

15. Computer apparatus according to claim 14, wherein the processor is arranged decrypt the encrypted data set a Tate or Weil pairing when operating on the first and second data sets and the third and fourth data sets.

16. Computer apparatus according to any of claims 1 to 7, wherein the cryptographic key is a signature key.

17. Computer apparatus according to claim 16, wherein the processor is arranged to sign a sixth data set with the signature key.

18. Computer apparatus according to claim 17, wherein the processor is arranged to sign the sixth data set with the signature key and a random number.

19. Computer apparatus according to claims 16 or 17, wherein the processor is arranged to sign the sixth data set using a bilinear pairing when operating on the first and second data sets and the third and fourth data sets.

20. Computer apparatus according to claim 19, wherein the bilinear pairing is either a Tate or Weil pairing.

21. Computer apparatus according to claims 1 to 7, wherein the cryptographic key is a verification key.

22. Computer apparatus according to claim 21, wherein the processor is arranged to verify a signed data set with the verification key.

23. Computer apparatus according to claim 21, wherein the processor is arranged to verify the signed data using a bilinear pairing when operating on the first and second data sets and the third and fourth data sets.

24. Computer apparatus according to claim 23, wherein the bilinear pairing is either a Tate or Weil pairing.

25. A method comprising generating a cryptographic key using a first data set that corresponds to a first identifier, a second data set corresponds to a first trusted party's public key, a third data set that corresponds to a second identifier and a fourth data set that corresponds to a second trusted party's public key.

26. A method according to claim 25, further comprising encrypting a fifth data set with the cryptographic key.

27. A method according to claim 26, wherein the fifth data set is encrypted using a Tate or Weil pairing when operating on the first and second data sets and the third and fourth data sets.

28. A computer system comprising a first computer entity arranged to generate a first data set that corresponds to a first trusted party's public key; a second computer entity arranged to generate a second data set that corresponds to a second trusted party's public key; and a third computer entity arranged to generate a cryptographic key using a first identifier in conjunction with the first data set and a second identifier in conjunction with the second data set.

29. A computer system according to claim 28, wherein the third computer entity is arranged to encrypt a third data set with the cryptographic key.

30. A computer system according to claim 29, wherein the third computer entity encrypts the third data set using a bilinear pairing when operating on the first and third data sets and the second and fourth data sets.

31. A computer system according to claim 30, wherein the bilinear pairing is either a Tate or Weil pairing.

32. A computer system according to claim 28, wherein the first data set and second data set are public data parameters.

33. A computer system according to claim 28, wherein the public data parameters include an elliptic curve and a generator point on the elliptic curve.

ABSTRACT

METHOD AND APPARATUS FOR GENERATING A CRYPTOGRAPHIC KEY

A computer system comprising a first computer entity arranged to generate a first data set; a second computer entity arranged to generate a second data set; and a third computer entity arranged to generate a cryptographic key using a first identifier in conjunction with the first data set and a second identifier in conjunction with the second data set.

Figure 1

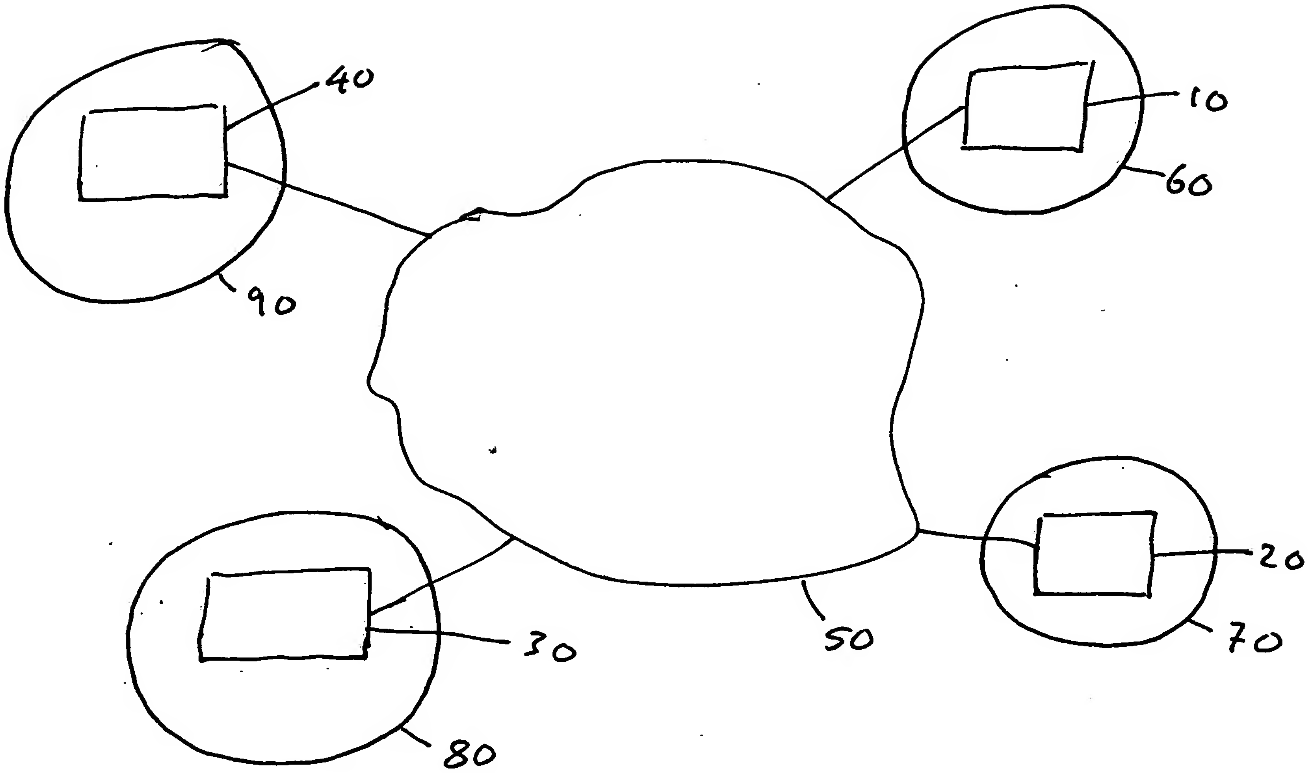

Figure 1

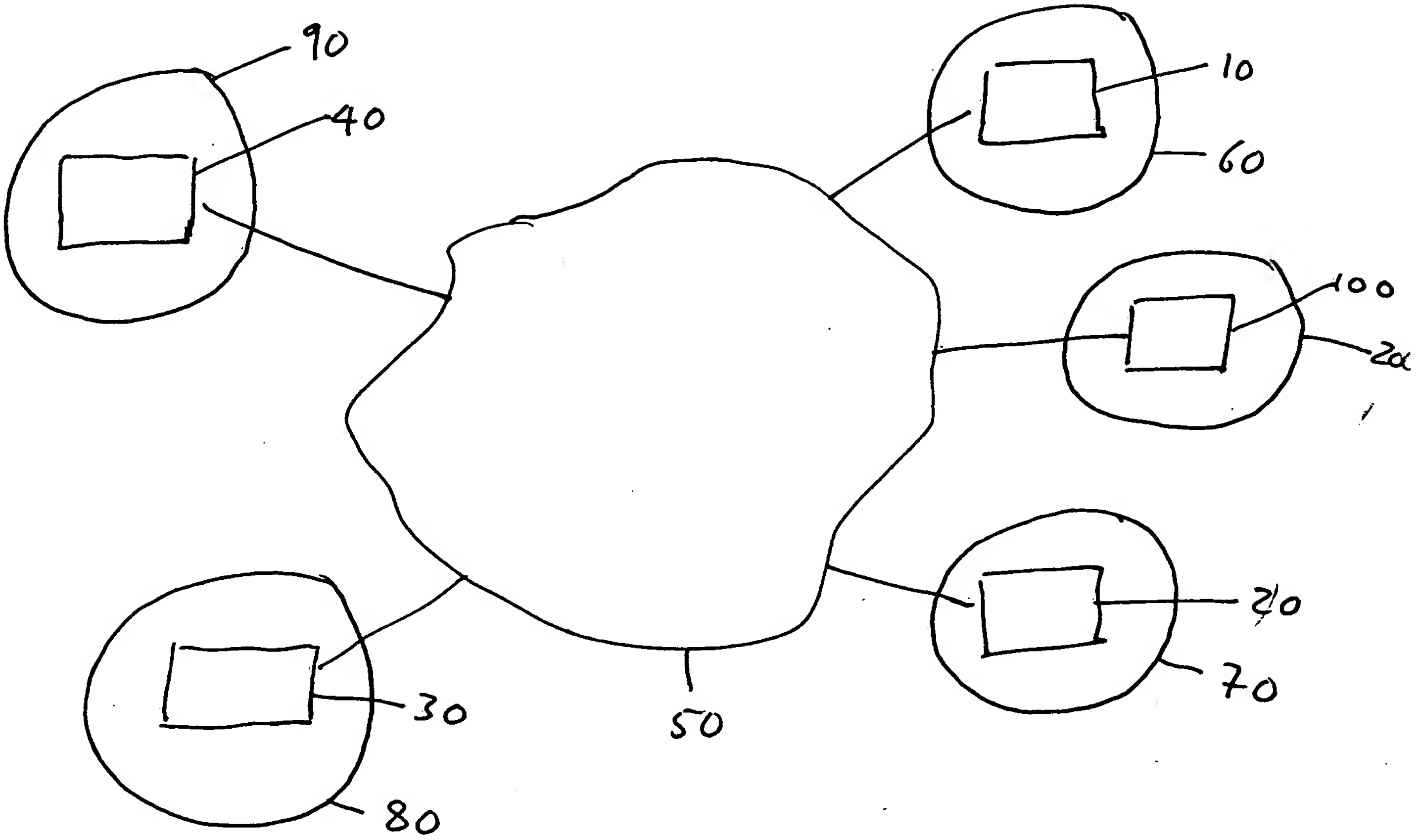

Figure 2